Homework 2

In this semester, the homework project let you apply the programming skills you learn in CSCI 4970/6970 Back-end Web Development by developing an application called SportsPro Technical Support. This application is designed for the technical support department of a hypothetical software company that develops software for sports leagues, and it uses a database named tech\_support.

The purpose of the application is to track technical support service calls (referred to as *incidents*) in a database that also stores information about the company’s customers, software products, and technicians.

The project contains two parts. Homework 1 is the first part of the project.

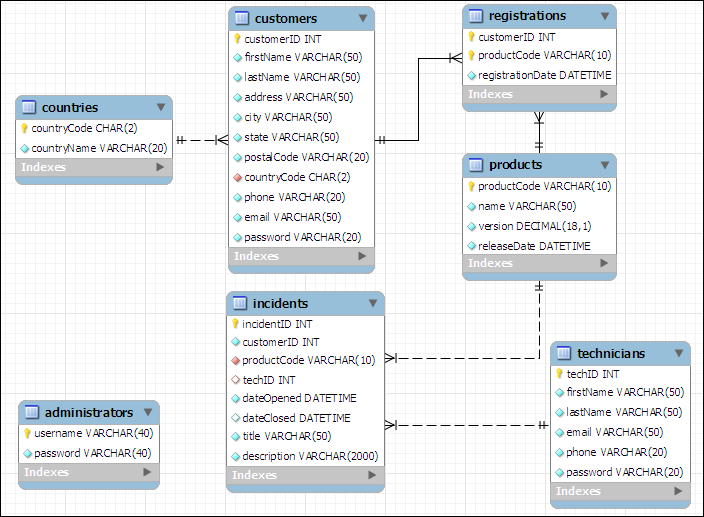
\*Note that this is not the class project.An introduction to the projects

This introduction describes the design of the SportsPro Technical Support application and the tech\_support database. In addition, it explains how to make the tech\_support database available to your applications, how to restore the database so it contains its original data, and how to prepare for developing the application. Finally, it provides some general information about developing the projects.

## The design of the tech\_support database

The tech\_support database is used to track technical support incidents. It consists of the seven tables shown in the diagram that follows. The incidents table contains one row for each technical support incident. Each row in the incidents table is related to one row in the customers table, which contains information about the company’s customers; one row in the products table, which contains information about the company’s products; and one row in the technicians table, which contains information about the company’s technical support staff.

In addition, a table named registrations keeps track of the products that are registered to each customer, a table named countries stores the countries of the world, and a table named administrators stores the usernames and passwords for the administrators. Note that the administrators table is not related to any of the other tables.



In addition to the column data types shown above, you should know that the customerID, incidentID, and techID columns in the customers, incidents, and technicians tables are AUTO\_INCREMENT columns. So, the values of these columns are set automatically when new rows are added to these tables. For more details about this database, you can use phpMyAdmin to view the structure and data that’s stored in the database.

## How to install the database

To install the tech\_support database, you can start phpMyAdmin or MySQL Workbench and run the tech\_support.sql file that’s provided by your instructor.

If you have already done that in homework 1. Just use the one that you created.

### How to restore the database

As you test some of the projects that you develop, you’ll need to add, modify, and delete rows in the database. Then, at some point, you may want to restore the original data. To do that, you can use phpMyAdmin to run the tech\_support.sql file again. This deletes both the structure and the data of the current tech\_support database and restores the original database.

### How to format the web pages

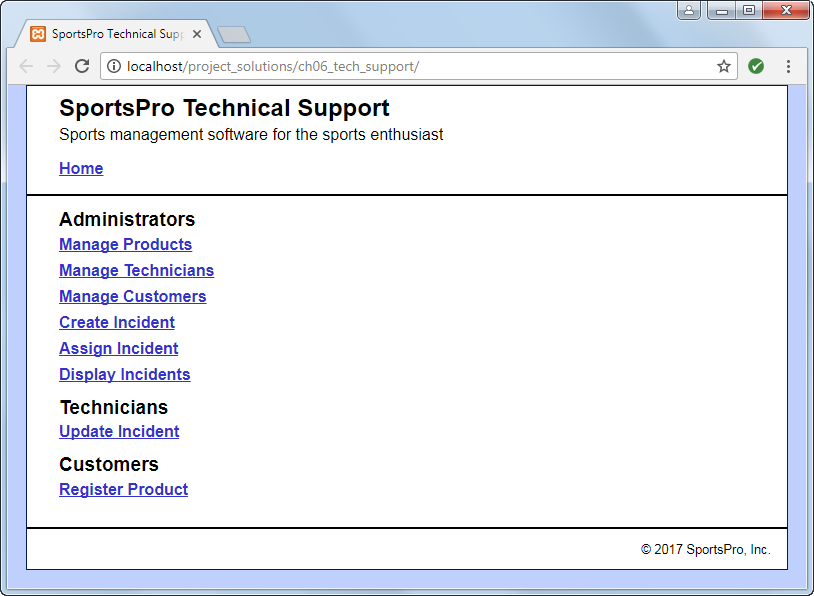
As you develop the web pages needed for each project, you will need to apply some formatting to them. To make that easier, you can use the main.css file that’s provided by your instructor. If necessary, you can modify this file, but it contains all of the tags needed to format the pages as shown in this document.

### What to create?

You must create 3 modules with MVC pattern (marked as Mandatory)

You can create the other two modules with MVC pattern to get the bonus points (marked as Optional/Bonus)

## A starting point for the projects



The starter project for homework 2 is the solution of homework 1. That means, you will keep working on what you should complete in homework 1.

# The projects

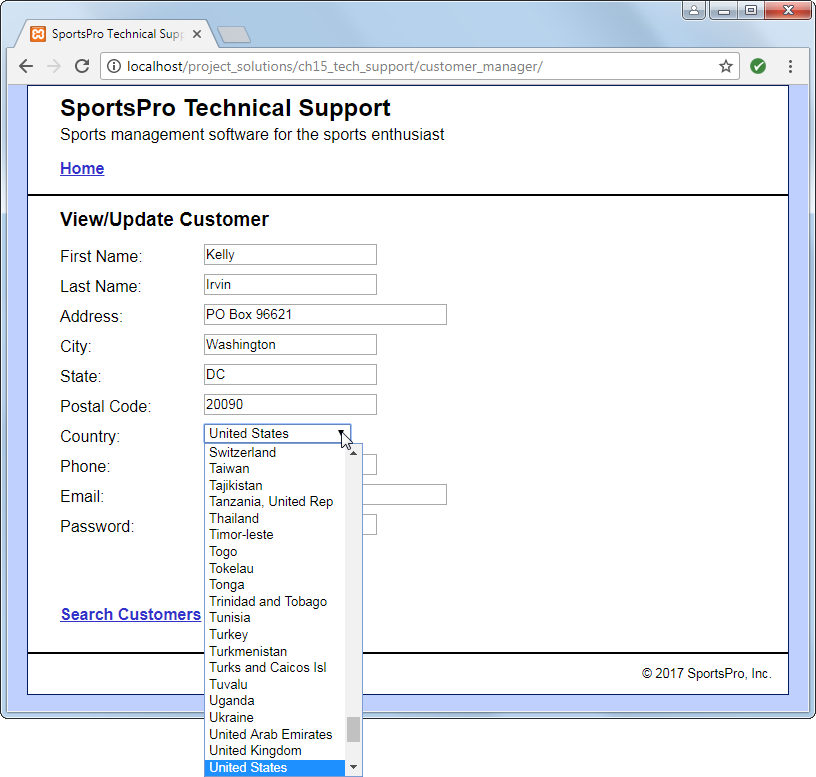
The description of each project includes an image of how the pages should appear in a browser, a description of how the pages operate, and specifications for how the project should be coded. This information is detailed enough for you to complete each project. However, you’ll need to use your best judgment on how to code many of the details. To do that, write the code in the way that you think is best, based on the skills that were presented in the book.

Unless you’re instructed otherwise, you can implement each project using any programming techniques you wish. In some cases, however, the project’s specifications will direct you to use a specific programming technique. For example, a project may direct you to use sessions. In that case, you should implement the project as directed.

Task 1 - Use a drop-down list

For this task, you’ll modify the Manager Customers application so it uses a drop-down list to display the country. Also, you’ll allow the user to use this drop-down to change the country. (*Required reading: chapters 1-7*)

The View/Update Customer page



Operation

* When the user selects a customer, the View/Update Customer page should display the country in a drop-down list, and it should select the correct country for the customer. If you have any trouble with this, look ahead to figure 8-11 in chapter 8.
* The user can use the Country drop-down list to change the country for the customer.

Specifications

* In the Country drop-down list, display all countries that are available in the countries table in the tech\_support database.
* When the page is first displayed, make sure to select the correct country for the specified customer. To do that, write code that sets the selected attribute of the <option> tag for the appropriate country.

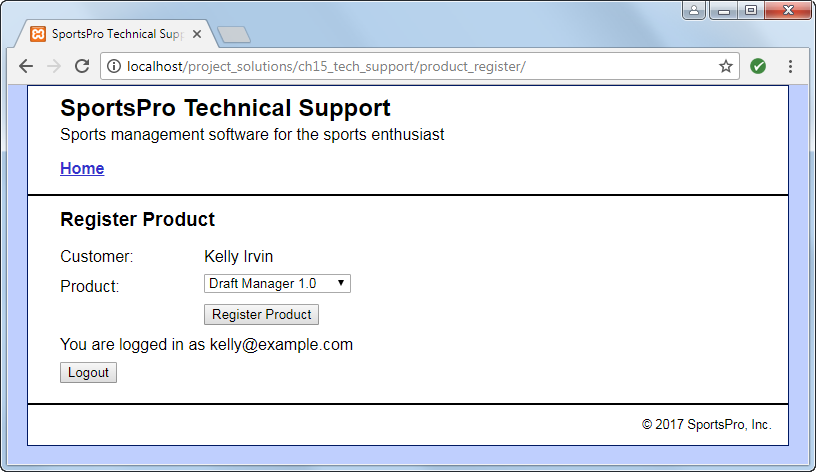
Task 2 - Improve controller code

For this task, modify one or more of the controller files so they use switch statements instead of if/else statements to select the appropriate action. (*Required reading: chapters 1-6 and 8*)

Task 3 - Use sessions

For the Register Product application that you created in homework 1, let the customer skip the Customer Login page if he or she has already logged in. (*Required reading: chapters 1-6 and 12*)

The Register Product page



Operation

* Same as Task 4 in homework 1, but a customer who has logged in can skip the Customer Login page.
* The customer can view the message on the Register Product page to verify that he or she is logged in.
* To log out, the customer can click on the Logout button or close the browser. If the user clicks the Logout button, the Customer Login page is displayed.

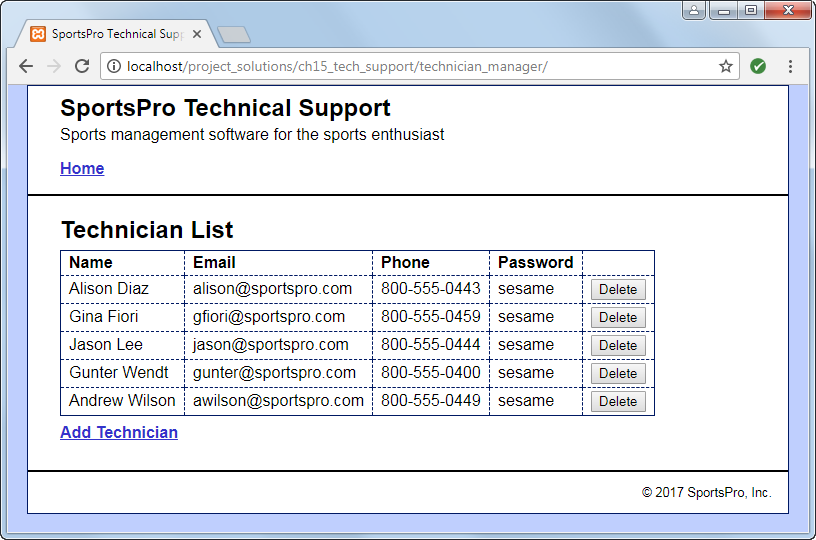
Specifications

* Use a session cookie for the session. That way, the session ends when the user closes the browser.
* Store the customer data in the session. That way, you don’t have to use hidden fields to pass the customer ID to the controller.

### Task 4 - Use objects

For the Manage Technicians in the starter project, use the object-oriented techniques described in chapter 14. (*Required reading: chapters 1-6 and 14*)

The Technician List page



Operation

* Same as Task 2 in Homework 1. However, for the Technician List page, the First Name and Last Name columns are combined into a single Name column that contains the full name.

Specifications

* Use a class named Database to get a connection to the database.
* Store the Database class in a file named database\_oo.php.
* Use a class named Technician to store data about each technician. This class should include a method that returns the full name of the technician.
* Store the Technician class in a file named technician.php.
* Use a class named TechnicianDB to store the methods that access the data. These methods should accept or return a Technician object or an array of Technician objects whenever appropriate.
* Store the TechnicianDB class in a file named technician\_db\_oo.php.

# Sumission

Option 1 – Zip your project folder that contains all necessary files and submit it to the blackboard.

Option 2 – Publish your website by using AWS EC2 or Beanstalk. Then submit the link to the blackboard (10 bonus points).